Extraction de commentaires utilisateurs sur le Web

Julien Subercaze, Christophe Gravier, Frederique Laforest

Extraction de commentaires utilisateurs sur le Web

Julien Subercaze*, Christophe Gravier*, Frédérique Laforest*

* Université de Lyon, F-42023, Saint-Etienne, France
CNRS, UMR5516, Laboratoire Hubert Curien, F-42000, Saint-Etienne, France,
Université de Saint-Etienne, Jean Monnet, F-42000, Saint-Etienne, France

Résumé. Dans cet article, nous présentons CommentsMiner, une solution d'extraction non supervisée pour l'extraction de commentaires utilisateurs. Notre approche se base sur une combinaison de techniques de fouille de sous-arbres fréquents, d'extraction de données et d'apprentissage de classement. Nos expérimentations montrent que CommentsMiner permet de résoudre le problème d'extraction de commentaires sur 84% d'un jeu de données représentatif et publiquement accessible, loin devant les techniques existantes d'extraction.

1 Introduction

Les grandes entreprises du Web ont depuis longtemps compris la valeur des contenus générés par leur utilisateurs – des services gratuits de grande qualité leur permettent d'atteindre plusieurs millions d'utilisateurs. Analyser et traiter ces données a fait émerger plusieurs défis scientifiques, il n'est donc pas surprenant de constater que ces contenus attisent les recherches en informatique. D'autres données sociales du même acabit, cette fois-ci disséminées sur la toile, sont les commentaires laissés par les utilisateurs du Web. Dans cet article, nous proposons `CommentsMiner`, un algorithme en deux temps permettant l'extraction des commentaires présents dans une page, ainsi que leur structure conversationnelle. Notre approche permet l'extraction de commentaires en réponse à des commentaires déjà publiés : l'extraction de la structure conversationnelle est fondamentale pour les tâches d'analyse en aval. Le problème d'extraction de commentaires utilisateurs peut être défini comme retrouver un motif p ayant été utilisé afin de générer le fragment HTML utilisé pour générer les commentaires dans la page. Ce motif est valide seulement à l'échelle d'un site Web : une fois le motif déterminé pour la page d'un site, la probabilité d'extraire correctement les commentaires présents dans les autres pages du site est grande.

La figure 1 présente l'architecture générale de `CommentsMiner`. Dans un premier temps, le problème d'extraction des commentaires est vu comme une tâche de fouille sous contraintes des sous-arbres fréquents induits et clos. Cette étape renvoie des motifs $p_i \in P$ récurrents dans la page (motif pour générer des commentaires, mais aussi les menus, publicités, contenus divers, etc.) et donc candidats pour être ceux utilisés pour générer le fragment HTML représentant les commentaires. Dans un second temps, l'apprentissage d'une fonction de classement permet de sélectionner le motif p étant le plus probablement celui utilisé pour présenter des commentaires dans la page. Nous présentons successivement ces étapes avant de présenter nos résultats expérimentaux dans la dernière section de cet article.

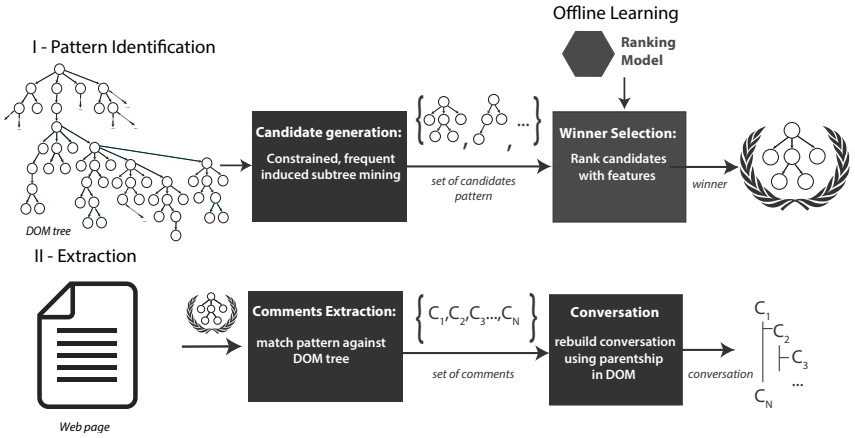

FIG. 1: Schéma de principe de CommentsMiner

2 Génération de candidats

Afin de réaliser la fouille de commentaires enfouis dans une structure conversationnelle, la fouille de sous-arbres fréquents doit permettre la suppression de feuilles horizontalement. Les sous-arbres visés sont donc induits en plus d'être fréquents.

Concernant les commentaires, leur contenu peut être du texte formaté à l'aide de tags HTML. Ces tags sont bien souvent laissés à la discrétion de l'utilisateur parmi une liste fermée et dépendant du site Web sur lequel les commentaires sont publiés. La présence de ces tags dans les commentaires utilisateurs empêche d'aborder le problème comme un problème de fouille de sous-arbres maximaux. Cepdendant, la fouille de sous-arbres clos satisfait notre cas d'utilisation puisque cette fouille identifiera les sous-arbres ciblés ainsi que leur sous-arbres parents – dans le cas où les commentaires ont des formatages différents, le sous-arbre cible aura une valeur de support plus grande que ses sous-arbres parents.

L'algorithme CMTreeMiner (Chi et al. (2005)) permet de fouiller les sous-arbres fréquents ordonnés induis et clos (da Jiménez et al. (2010)). Nous réalisons cependant une fouille de tels sous-arbres sous contraintes. Nous avons donc adapté cette algorithmes à ces contraintes, que nous présentons dans la section suivante.

2.1 Contraintes

L'espace de recherche d'un arbre DOM est très large, puisqu'un arbre DOM contient en moyenne 1300 nœuds [1]. Même si CMTTreeMiner est l'algorithme le plus performant pour cela, les auteurs conviennent que la fouille d'un millier de nœuds avec un valeur support de 2 peut s'approcher d'une heure. Dans le cas spécifique des commentaires, nous pouvons cependant identifier trois contraintes spécifiques à ce domaine afin de réduire l'espace de recherche.

1. http://www.httparchive.org/trends.php?s=Top1000

Similarité du plus proche ancêtre commun. Comme les commentaires sont dans des zones uniques de l'arbre DOM, les occurrences du motif cible sont relativement proche les unes des autres. Autrement dit, le plus proche ancêtre commun entre deux occurrences du motif cible ne peut être la racine de l'arbre DOM.
Suppression des occurrences vides. `CommentsMiner` ignore les motifs cibles candidats dont les occurrences sont « blanches », i.e. elles ne contiennent aucun texte ou le même texte partagé entre toutes les occurrences.
Unicité des racines et occurrences la plus à droite. Notons $RootOcc_{t,T}$ et $RmoOcc_{t,T}$ les ensembles de racines et occurences les plus à droite d'un sous-arbre fréquent t dans un arbre T. Pour les commentaires, chaque occurence de t a sa propre racine et sa propre occurence la plus à droite – lesquels ne sont pas partagés avec un autre commentaire. Une contrainte d'intégrité sur t est donc vérifiée à l'aide de l'égalité suivante : $|RootOcc_{t,T}| = |RmoOcc_{t,T}|$. Si un motif candidat viole cette contrainte, `CommentsMiner` se défausse de ce candidat.

Cette étape de `CommentsMiner` génère un ensemble de motifs candidats, parmi lesquels se trouve le motif utilisé pour injecter dans la page Web des fragments HTML correspondant aux commentaires. Les contraintes décrites dans cette section permettent de limiter drastiquement le nombre de motifs candidats. En pratique, ce nombre dépasse rarement vingt.

3 Sélection du motif cible parmi les motifs candidats

Identifier le motif p parmi l'ensemble des motifs candidats peut être appréhendé comme un problème d'apprentissage d'une fonction de classement, où seul la précision au premier rang importe. Afin de classifier ces motifs candidats, `CommentsMiner` utilise des descripteurs textuels et densitometriques utilisés par des algorithmes d'apprentissage de classement : SVMRank, MART, RankNet, RankBoost, AdaRank, Random Forests, et la programmation génétique. Nous décrivons ces descripteurs avant de présenter les résultats expérimentaux.

Les contenus générés par les utilisateurs sont de taille variable (Kohlschütter et Nejdl (2008)) – à l'inverse des menus ou publicités pour lesquels la taille et le nombre de mots tendent à être similaires entre leurs différentes occurrences. Il est également important de se rappeler qu'il est souvent interdit d'inclure des liens hypertextes dans les commentaires afin de limiter les contenus indésirables. Nous pouvons donc pénaliser les motifs candidats ayant un trop grand nombre de liens en fonction de la taille du commentaire. De même, les commentaires peuvent se discriminer également de part leur ratio de texte sur code HTML faible, à l'inverse des autres contenus (Agichtein et al. (2008)). La moyenne ainsi que le coefficient de variation représentent de manière latente l'hétérogénéité entre les occurrences d'un motif candidat. `CommentsMiner` exploite ces observations sous la forme de huit descripteurs (moyenne et coefficient de variation pour la densité de texte, la taille du texte, la densité de liens, la taille du texte, ainsi que la volumétrie mesurée en nombre de mots).

Notons que `CommentsMiner` repose sur une fonction de classement qui doit nécessairement être apprise. Cependant, le système peut être appréhendé comme non supervisé dans certaine classification du domaine : une fois la fonction apprise, elle peut être réutilisée pour d'autres domaines Web, sans connaissance a priori de ce domaine ou de la structure de l'arbre DOM. Cela est consistent avec la classification de l'état de l'art exhaustif publié récemment (Sleiman et Corchuelo (2013)), même si cela demeure discutable.

À l'exécution, une fois le motif cible identifié, `CommentsMiner` peut procéder à l'extraction de commentaires. Comme décrit à la figure 1, le système sélectionne les occurrences du motif cible identifié et reconstruit la structure conversationnelle des commentaires en fonction de l'emplacement de ses occurrences dans l'arbre DOM. La sélection des occurrences du motif est réalisé en temps linéaire à l'aide d'une stratégie de parcours en profondeur de l'arbre DOM.

4 Résulats expérimentaux

Cette partie présente les jeux de données et les algorithmes utilisés comme points de comparaison. La performance et l'exactitude de `CommentsMiner` pour l'extraction de commentaires sont ensuite reportées et discutées.

Points de comparaison. À notre connaissance, seul MiBat (Song et al. (2010)) a été conçu pour la tâche d'extraction de commentaires. MiBat n'est pas reproductible : les jeux de données ou implémentations ne sont pas accessibles publiquement, ni sur demande. MiBat présente un taux de succès de 75.653% pour différentes pages Web appartenant au même domaine Web – cela est déduit de l'illustration présente dans l'article, même si leur nombre et leur contenu demeurent inconnus – ce qui biaise l'évaluation. Un autre point de comparaison possible est DEPTA (Zhai et Liu (2005)), une suite donné aux travaux de l'extracteur de région MDR (Liu et al. (2003)). DEPTA nécessite le rendu complet de la page côté navigateur, et une analyse basée sur un traitement d'image limite sa capacité à passer à l'échelle. De plus, DEPTA n'est pas capable d'extraire les relations parent-enfant pour les occurrences des motifs qu'il identifie. DEPTA n'est pas accessible non plus publiquement, mais son prédécesseur MDR est accessible en ligne. Cela pose MDR comme un point de comparaison de choix dans de très nombreux travaux, comme rapporté dans l'enquête menée par Sleiman (Sleiman et Corchuelo (2013)). TPC (Miao et al. (2009)) et RST (Bing et al. (2011)) sont d'autres points de comparaison possibles. Ils ne sont pas non plus accessibles – il est possible que la valuation de la donnée extraite, des commentaires, inaccessible autrement, contribue à expliquer cette tendance. Ils ont néanmoins été testé sur le même jeu de données, à savoir le jeux de données TestBed for information extraction from Deep Web (TBDW) [2]. `CommentsMiner` atteint un score parfait de 100% sur ce jeu de donnée, ce qui fait en réalité peu de différence avec TPC et RST (respectivement une précision de 96.23% et 98.06% avec un rappel respectif de 97.03% et 97.88%).

Le jeu de données NUCE. En plus de la faible discrimination entre les systèmes existants fourni par le TBDW, ce jeu de données ne reflète plus les habitudes de programmation Web contemporaines. Plus spécifiquement, *(i)* les balises `<table />` et `<form />` ne sont plus utilisées actuellement pour organiser le placement d'information sur une page Web, *(ii)* le jeu de données ne contient pas de commentaires en réponse à des commentaires, *(iii)* la taille d'une page Web a augmentée de 237% entre décembre 2010 [3] et février 2015 [4] – ce qui a pour effet d'augmenter exponentiellement l'espace de recherche de sous-arbres représentant des motifs candidats, ce que le jeu de donnée ne reflète pas. Nous avons donc construit un nouveau jeu de données avec i) des pages Web contemporaines, ii) issues de domaines Web multilingues et

2. `http://daisen.cc.kyushu-u.ac.jp/TBDW/`
3. `http://httparchive.org/interesting.php?l=Dec%2028%202010`
4. `http://httparchive.org/interesting.php?l=Dec%2015%202013`

uniques (aucune page issue du même domaine), iii) des pages Web contenant des commentaires enfouis dans les commentaires auquel ceux-ci répondent. À partir de Google News anglais, français et allemand, nous avons sauvegardé une page contenant au moins deux commentaires par domaine. Le contenu de cette page a été téléchargée à l'aide d'un navigateur afin d'éviter les problèmes d'appels AJAX Gyllstrom et al. (2012). Il existe des services d'hébergement de commentaires dans le cloud, tels que Wordpress, Disqus, Livefyre, Facebook, etc. Pour éviter tout biais de duplication de motif cible, nous n'avons conservé qu'une seule page faisant appel à de tels services. Ce jeu de données, appelé NUCE pour Nested User-generated Content Extraction, contient 211 pages labélisées. Ce jeu de données de remplacement est accessible publiquement, et inclut pour chaque page l'arbre DOM tel que fournit par le navigateur après rendu mais également la vérité de terrain quant au motif cible (sous-arbre) à identifier.

Resultats L'évaluation de notre système dépend principalement de l'étape d'apprentissage d'une fonction de classement puisque que le motif p est toujours inclut dans l'ensemble des motifs candidats (comme expliqué à la Section 2)). Nous utilisons différentes approches pour l'apprentissage d'une fonction de classement. Les travaux de Cao et al. (2007) et Xia et al. (2008) fournissent par ailleurs un état de l'art complet sur ces approches d'apprentissage de fonctions de classement. Les programmes génétiques ont été entrainés à l'aide des opérateurs suivants : addition, multiplication, soustraction, division, fonction puissance ainsi que les valeurs réelles dans l'intervalle $[2; 10,000]$. Tous les modèles ont été entrainés en utilisant les huit descripteurs définis précédemment. Les résulats sont présentés dans le tableau 1. L'entrainement est réalisé sur 20% du jeu de données et une validation croisée est réalisée cinq fois. ListNet fournit le modèle le plus efficace avec une précision au rang 1 (P@1) de 90.170 sur 100 exécutions. Néanmoins, nous pensons que les modèles à base de programmation génétique sont donc plus adaptés : bien que ces modèles n'atteignent pas le plus haut taux de succès (84.375 en moyenne), ils présente une bonne robustesse à des fins de généralisation (coefficient de variation de 0.854).

Algorithm	Settings	*Mean*	*STD*
MART	$1,000\ trees$	66.4	6.3
SVMRank	$RBF,\ c = 0.1$	77.1	6.3
RankNet	$100\ itrations$	67.1	6.8
RankBoost	$300\ tours$	84	4.9
AdaRank	WTA pour l'apprentissage	66	15.4
Coord. Ascent	WTA pour l'apprentissage	81.6	3.6
ListNet	$1,500\ itrations$	90.1	13.5
Gen. Prog.	$\mathbf{50\ itrations}$	**84.3**	**0.8**
MiBAT (*baseline*)	N/A	75.6	Inconnu

TAB. 1: Performance et configuration pour les modèles appris sur le jeu de données NUCE.

5 Conclusions et travaux futurs

Nous avons présenté `CommentsMiner`, une nouvelle approche pour extraire des commentaires. `CommentsMiner` exploite des contraintes pour la fouille de sous-arbres fréquents, clos et induits, afin d'extraire les commentaires publiés en réponse à un commentaire existant. L'exploitation de contraintes rendue possible par le cadre applicatif considéré permet de réduire drastiquement l'espace de recherche et d'éviter l'habituelle explosion combinatoire lors de fouille de sous-arbres fréquents. Un apprentissage de fonction de classement permet ensuite d'identifier le motif cible parmi ceux étant fréquents. Nous avons également démontré les performances de cette approche pour un jeu de données populaires (quoique désuet) mais également un jeu de données supplémentaire que nous proposons à la communauté.

Références

Agichtein, E., C. Castillo, D. Donato, A. Gionis, et G. Mishne (2008). Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 183–194. ACM.

Bing, L., W. Lam, et Y. Gu (2011). Towards a unified solution : data record region detection and segmentation. In *Proc. of the 20th CIKM Conference*, pp. 1265–1274.

Cao, Z., T. Qin, T.-Y. Liu, M.-F. Tsai, et H. Li (2007). Learning to rank : from pairwise approach to listwise approach. In *Proc. of the 24th ICML Conference*, pp. 129–136.

Chi, Y., Y. Xia, Y. Yang, et R. R. Muntz (2005). Mining closed and maximal frequent subtrees from databases of labeled rooted trees. *Knowledge and Data Engineering, IEEE Transactions on 17*(2), 190–202.

da Jiménez, A., F. Berzal, et J.-C. Cubero (2010). Frequent tree pattern mining : A survey. *Intell. Data Anal. 14*(6), 603–622.

Gyllstrom, K., C. Eickhoff, A. P. de Vries, et M.-F. Moens (2012). The downside of markup : examining the harmful effects of css and javascript on indexing today's web. In *Proc. of the 21st CIKM Conference*, pp. 1990–1994.

Kohlschütter, C. et W. Nejdl (2008). A densitometric approach to web page segmentation. In *Proc. of the 17th ACM CIKM Conference*, pp. 1173–1182.

Liu, B., R. Grossman, et Y. Zhai (2003). Mining data records in web pages. In *Proc. of the 9th KDD Conference*, pp. 601–606.

Miao, G., J. Tatemura, W.-P. Hsiung, A. Sawires, et L. E. Moser (2009). Extracting data records from the web using tag path clustering. In *Proc. of WWW*, pp. 981–990.

Sleiman, H. et R. Corchuelo (2013). A survey on region extractors from web documents. *IEEE trans. on Knowledge and Data Engineering 25*(9), 1960–1981.

Song, X., J. Liu, Y. Cao, C.-Y. Lin, et H.-W. Hon (2010). Automatic extraction of web data records containing user-generated content. In *Proc. of CIKM*, pp. 39–48.

Xia, F., T.-Y. Liu, J. Wang, W. Zhang, et H. Li (2008). Listwise approach to learning to rank : theory and algorithm. In *Proc. of the 25th ICML Conference*, pp. 1192–1199.

Zhai, Y. et B. Liu (2005). Web data extraction based on partial tree alignment. In *Proc. of the 14th WWW Conference*, pp. 76–85.

Summary

We present `CommentsMiner`, an user-generated comments extractor. CommentsMiner combines frequent closed induced subtree mining and a learning-to-rank model. It mainly relies on textual and densitometric to mine and train its model. We show that `CommentsMiner` perfectly solves the comments extraction task with a 84% performance on a representative and publicy available dataset – another technical contribution of this paper.